

UNITED STATES PATENT APPLICATION
FOR
AUTOMATIC READ OF CURRENT TIME WHEN EXITING LOW-POWER
STATE

INVENTORS:

David I. Poisner

Prepared by:

BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP
32400 Wilshire Boulevard
Los Angeles, CA 90025-1026
(408) 720-8300

Attorney's Docket No.: 42390P12974

"Express Mail" mailing label number: EL431887670US

Date of Deposit: January 2, 2002

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to the Commissioner for Patents, Washington, D. C. 20231

Carla Zavala

(Typed or printed name of person mailing paper or fee)

(Signature of person mailing paper or fee)

(Date signed)

1-202

AUTOMATIC READ OF CURRENT TIME WHEN EXITING LOW-POWER STATE UTILITY

FIELD OF THE INVENTION

[0001] This invention relates to the field of power management of a computer system processor; more particularly, to the computation of a duration of a reduced power consumption state.

BACKGROUND

[0002] Most of the computer systems have power management tools incorporated within, that allow the system's processor to conserve power by entering power safe modes. Due to a variety of power safe modes it is essential that an operating system be able to intelligently determine which power safe mode to enter at a particular time. Some operating systems, especially those that support Advanced Configuration and Power Interface (ACPI), details of which may be found in Advanced Configuration and Power Interface Specification, Rev. 2.0 (July 27, 2000), attempt to measure the duration of time that the Central Processing Unit (CPU) has been placed in power states, such as C1, C2, C3 , etc. Depending on the duration of the low-power state the operating system may make an intelligent selection of a low-power state in the future. For example, if the CPU enters a C3 state and remains in the state only for a short time, the operating system may in the future decide to enter C2 state rather than C3 state. On the other hand, if the CPU enters C2 power state for a long time, the operating system may select a C3 state next time it needs to enter a low-power state.

[0003] In order to calculate the duration of a particular power state, the operating system needs to determine the time at which the CPU entered the low-power state and the time at which the CPU resumed execution of instructions.

[0004] For power states, such as C2 and C3, the processor initiates the transition to the state by accessing a register in a chipset, which halts the processor. Upon exiting the low-power state, the CPU starts the execution at an instruction next to the one that it completed executing prior to entering the low-power state. Thus, the operating system, is able to determine the entering and exiting time of the power state prior to resuming the execution of instructions.

[0005] However, the C1 power state is entered by the CPU executing a halt instruction, not by accessing a register. Exit from the C1 low-power state is interrupt driven and the CPU jumps directly to an interrupt service routine, which may be message based, pin based or any other type of interrupt routines. Thus, in a case of the C1 power state the operating system cannot directly measure the duration of time that the CPU was in the C1 power state because the first instruction that is executed does not result in the exit time being stored. The operating system can estimate the duration of the C1 power state, and in a case of an inaccurate estimation the selection of the next low-power state will reflect the inaccuracy.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] The present invention will be understood more fully from the detailed description given below and from the accompanying drawings of various embodiments of the invention, which, however, should not be taken to limit the invention to the specific embodiments, but are for explanation and understanding only.

[0007] Figure 1 illustrates one embodiment of a computer system with an exit time register being located in a CPU;

[0008] Figure 2A is a flow diagram illustrating one embodiment of a process for storing current time prior to entering a power state;

[0009] Figure 2B is a flow diagram illustrating one embodiment of a process for storing current time upon exiting the power state;

[0010] Figure 3 illustrates one embodiment of a computer system with exit time register located in a chipset;

[0011] Figure 4 illustrates one embodiment of a computer system with an entrance time register and the exit time register located in a chipset;

[0012] Figure 5A is a flow diagram illustrating one embodiment of a process for requesting a counter controller to start a time counter;

[0013] Figure 5B is a flow diagram illustrating one embodiment of a process for requesting the counter controller to halt the time counter; and

[0014] Figure 6 illustrates one embodiment of a computer system with a time counter located in the chipset.

DETAILED DESCRIPTION

[0015] In the following description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that these specific details need not be employed to practice the present invention. In other instances, well known materials or methods have not been described in detail in order to avoid unnecessarily obscuring of the present invention.

[0016] A method and apparatus for computing a duration of a power management state of a computer processor is described. In one embodiment, a method is described for computing a C1 power state duration by recording the time of the processor's entrance into the C1 power state and the time of the processor's exit from the C1 power state.

Power-Management Technology

[0017] As indicated above, a method and apparatus for computing a duration of a power management state of a computer processor is described. Accordingly, some introduction to processor's power management technology is helpful in understanding the teachings described herein.

[0018] Embodiments described herein may utilize operating systems that support Advanced Configuration and Power Interface (ACPI). ACPI is a power management specification that allows hardware status information to be available to an operating system. ACPI allows a Personal Computer (PC) to turn

its peripherals on and off for improved power management. In addition, ACPI allows the PC to be turned on and off by external devices, such as a mouse or a keyboard.

[0019] ACPI supports several processor power states, including a C1 power state. When in C1 power state the processor consumes less power and dissipates less heat than in an active power state where the CPU executes instructions. The C1 power state is supported through active instructions of the processor, without requirement of additional hardware support, and it allows the processor to maintain the context of the system caches. The C1 state is exited when an interrupt is presented to the processor.

Methodology

[0020] With these concepts in mind exemplary embodiments may be further explored. Figure 1 illustrates one embodiment of a computer system. Prior to a CPU 100 entering the C1 power state, the operating system 130 reads the current time at 200 of Figure 2A and stores the current time in an entrance time register 120 located in a main memory 115. In one embodiment the operating system 130 reads the current time maintained by a hardware timer, for example an 8254 timer. Upon storing the current time in the entrance time register 120, the operating system 130 executes a HALT instruction. The HALT instruction is an instruction, execution of which, causes the processor to stop executing instructions. In one embodiment the entrance into the C1 power state

may be performed by setting a control bit in the processor instead of the execution of the HALT instruction. It will be appreciated that there may be other methods of entering into the C1 power state.

[0021] In one embodiment, upon an interrupt being presented to the processor and prior to the processor exiting the C1 power state, the operating system 130 at 215 of Figure 2 reads the current time and at 220 stores the current time in an exit time register 105 of Figure 1. It will be appreciated that the invention is not limited to storing the current time on the processor and may be stored anywhere, for example in main memory. Upon storing the current time in the exit time register 105, the operating system 130 at 225 allows an interrupt routine to execute. At 230 of Figure 2 when the interrupt routine completes its execution, the operating system 130 computes the duration of the C1 power state. In one embodiment, the operating system 130 reads the values of the exit time register 105 and the entrance time register 120 and computes the time difference in order to obtain the C1 power state duration.

[0022] In one embodiment, prior to the execution of the interrupt routine, the operating system 330 sends the exit time to a chipset 310 of Figure 3 for storage thereon. This may involve performing a cycle to the chipset 310 and requesting storage of the current time in the exit time register 305. It will be appreciated that the storing of the current time may be performed by utilizing latches, flip-flops or other elements and techniques well known in the art. In addition, it will be appreciated that the cycle performed by the operating system

330 is not limited to any particular cycle and maybe be an I/O cycle, memory cycle, or any other cycle type. Moreover, any type of a chip may be utilized to store the exit time.

[0023] Figure 4 illustrates an alternate embodiment of a computer system in which, prior to the execution of the HALT instruction, the operating system 430 requests the chipset 410 to store the current time in the entrance time register 420. When an interrupt is presented to the processor, the operating system 430 requests the chipset 410 to store the current time in the exit time register 405. Upon receiving the second request, a subtractor 425 located in the chipset 410 automatically performs a subtraction of values of the exit time register 405 and entrance time register 420 to determine the C1 power state duration.

[0024] According to another embodiment of a computer system, at 500 of Figure 5A the operating system 630 of Figure 6 requests a counter controller 605 located in the chipset 610 to start a time counter 620 prior to executing the HALT instruction. At 510 the operating system 630 executes the HALT instruction. When the interrupt is presented to the processor, the operating system 630 at 515 requests the counter controller 605 to halt the time counter 620. In this embodiment the C1 power state duration may be obtained by reading the contents of the time counter 620.

[0025] It will be appreciated that the above-described system and method may be utilized in a multiprocessor systems. In addition, it will be noted that the

terms 'CPU' and 'processor' are used interchangeably in the above description of the invention.

[0026] Whereas many alterations and modifications of the present invention will no doubt become apparent to a person of ordinary skill in the art after having read the foregoing description, it is to be understood that any particular embodiment shown and described by way of illustration is in no way intended to be considered limiting. Therefore, references to details of various embodiments are not intended to limit the scope of the claims, which in themselves recite only those features regarded as essential to the invention.